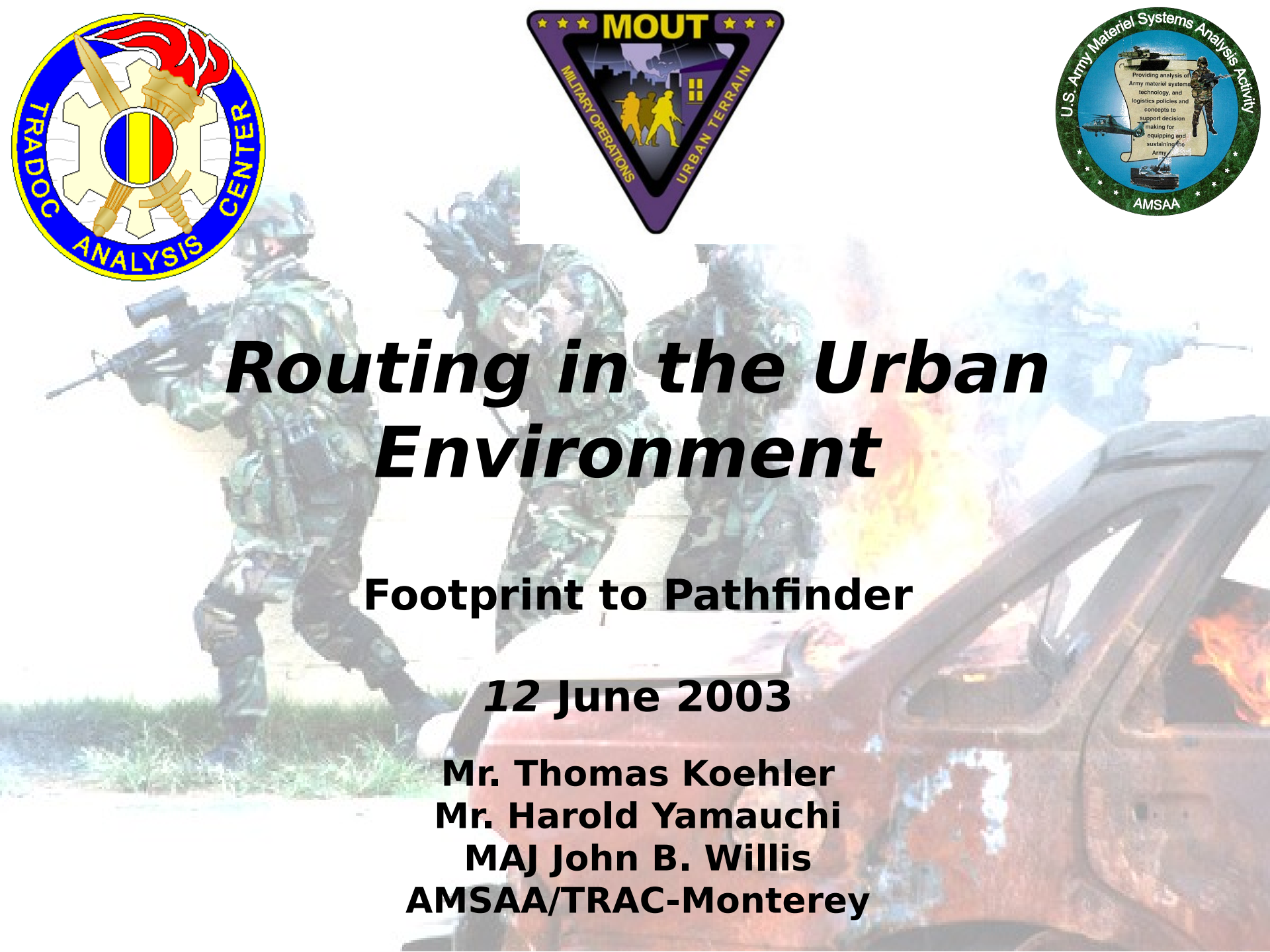


Routing in the Urban Environment

Footprint to Pathfinder

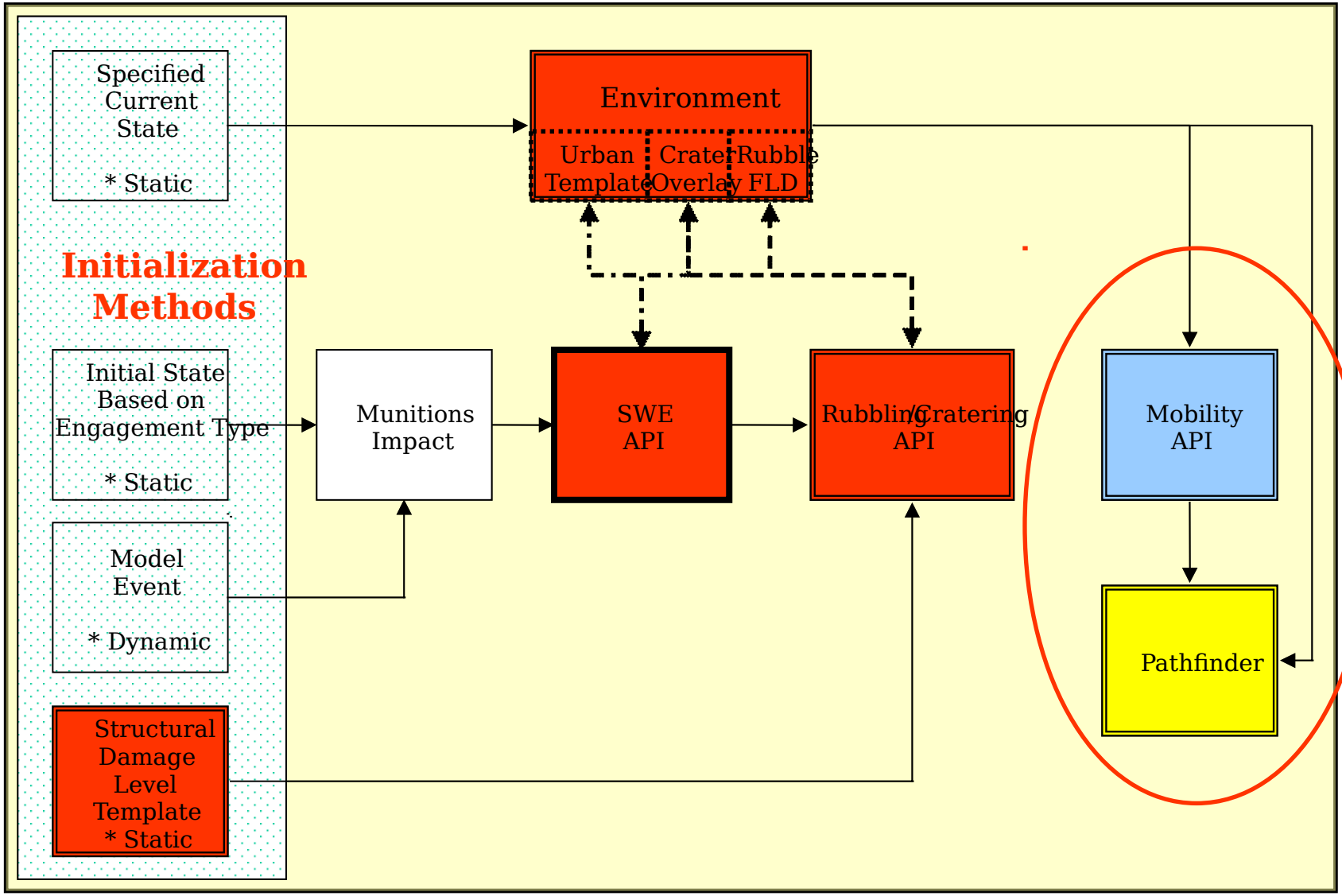
***12* June 2003**

Mr. Thomas Koehler
Mr. Harold Yamauchi
MAJ John B. Willis
AMSAA/TRAC-Monterey

Footprint to Pathfinder Overview

- **The US Military's involvement in urban operations has escalated significantly over recent years (Panama, Kuwait City, Haiti, Somalia, Bosnia, Kosovo, etc.).**
- **Use of modeling and simulation will aid in our assessment of our ability to fight in urban environments.**
- **These urban environments will consist of a dynamic terrain that will be characterized by the presence of complex obstacles, e.g., rubble piles and craters.**
- **Our work focuses on developing a methodology for determining optimal routes through urban road terrain networks, with careful consideration given to the dynamic urban environment.**

API Schematic: Model Integration



Mobility API Representative Vehicles

Vehicle	OOS/WARSIM Name	CCTT-SAF
M1A1	High Mobility Tracked	High Mobility Tracked
M270 MLRS	Medium Mobility Tracked	Good Mobility Tracked
M60 AVLB	Low Mobility Tracked	Low Mobility Tracked
M1084 MTV	High Mobility Wheeled	High Mobility Wheeled
M985 HEMTT	Medium Mobility Wheeled	Low Mobility Wheeled
M917 Dump Truck	Low Mobility Wheeled	Not Applicable
M1084/M1094	High Mobility Wheeled w/Towed Trailer	Not Applicable
M985/M989	Medium Mobility Wheeled w/Towed Trailer	Not Applicable
M911/M747 HET	Low Mobility Wheeled w/Towed Trailer	Not Applicable
M113A2	Tracked ACV	Moderate Mobility Tracked
LAV25	Wheeled ACV	Not Applicable
Kawasaki ATV	Light ATV	Not Applicable

Pathfinder Overview

Problem: Routing vehicles in an urban environment

- **Initial concentration of effort: Since urban areas are dominated by roads with a few open areas, let's look at solving the routing problem on the road network first**
- **Future work will consider bypass and open areas such as parking lots.**

Solution: Model road networks with an arc-node network structure

- **Numerous routing algorithms for arc-node networks are available in the open literature.**
- **Develop**

  1) an arc-node network generator,

  2) a route planning service to solve the shortest path problem for individual vehicles.

- **Programming language: Java (due to initial integration with Combat XXI)**

How do you generate an Arc-Node Network?

- **General:**
  - **Read the road network from an existing terrain database.**
  - **We currently are reading from CTDB using the ModTerrain API.**
  - **Additional sources for the future:**
    1) **Environmental Runtime Component (ERC)**
    2) **Geotypical Urban Templates**
    3) **Others?**

Shortest Path Problem

- **We are given a weighted, directed graph G = (V, E) where V is the set of nodes and E is the set of arcs. Each arc e_{ij} from node *i* to node *j* has a real-valued, non-negative cost c_{ij}.**
- **Find the shortest path from source node *s* to terminus node *t* by solving:**

$$\text{Min} \quad \sum_{j}^{N} \sum_{i}^{N} C_{ij} X_{i}$$

$$\text{Subject to} \sum_{j}^{N} X_{ij} - \sum_{j}^{N} X_{ji} = S_{i} \; \forall i$$

Where N = total number of nodes,

i, j = nodes,

C_{ij} = the cost of travel to node j from node i (≥ 0),

X_{ij} = 1 if i and j are on the shortest path, 0 otherwise,

S_{i} = 1 if i = s, -1 if i = t, 0 otherwise.

Shortest Path Algorithms

Dijkstra's Algorithm

- Well known and documented algorithm
- The algorithm has been studied with various heap (or priority queue) data structures to enhance run-time performance. (We use a min-priority queue.)
- Both algorithm and underlying data structure are easy to implement.
- Can be applied as long as all arc costs are nonnegative.

A* Algorithm

- Basic idea: use the geometric information inherent in the road network that is ignored by Dijkstra's algorithm.
- Finds shortest path from source *s* to destination *t* using a heuristic estimate h(*i*) ≥ 0 measured from node *i* to *t*. (Dijkstra's algorithm is the A* algorithm with h(*i*) = 0.)
- The estimate is not larger than the actual shortest path to *t*.
- A* runs faster than Dijkstra because the former searches in the direction of *t* while the latter searches in all directions regardless of where *t* is.
- When to apply: when arc costs are proportional to the Euclidean distance between the nodes.

Example Arc Cost for Combat^{XXI}

- **Cost currently considers two factors: travel time and risk from perceived threats**
- **Travel time is a function of speed and distance.**

  1) Distance is obtained from the arc's length.

  2) Speed is obtained from a table of road speeds provided by the Standard Mobility API (stored in a Microsoft Access database file).
- **The risk is obtained from "threat circles" laid over the network.**
- **Total arc cost:**

  $C_{ij} = (W_{travel} * T_{ij}) + (W_{risk} * R_{ij})$

  $0 < W_{travel} \leq 1$

  $0 \leq W_{risk} < 1$

  $W_{travel} + W_{risk} = 1$

  T_{ij} = travel time component (normalized)

  R_{ij} = risk component (scale from 1 to 100)

Arc Costs – Other Factors

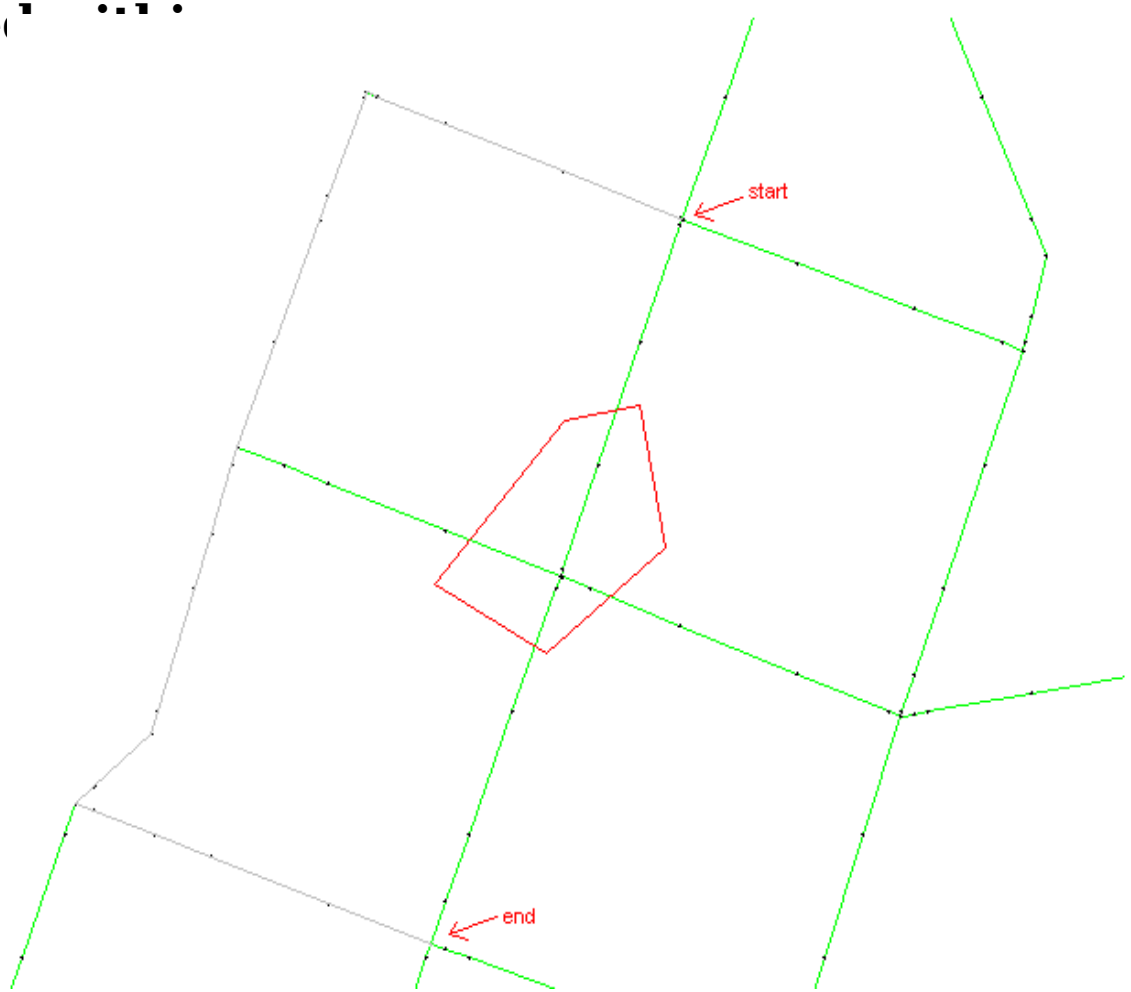
- **Cost function is user-defined**
- **Other factors that may be considered for arc cost:**
  1) **distance**
  2) **slope change**
  3) **number of turns**
  4) **surface type change**
  5) **cover/concealment/exposure**
  6) **river crossings**

Additional Constraints

- *No go area*
- *Axis of advance*
- *Threat circle*
- **No later than/not earlier than time arrival at destination**
- *Time delay at node caused by turning*
- **Rubble**
- **Turn restriction/Detour**

No Go Area

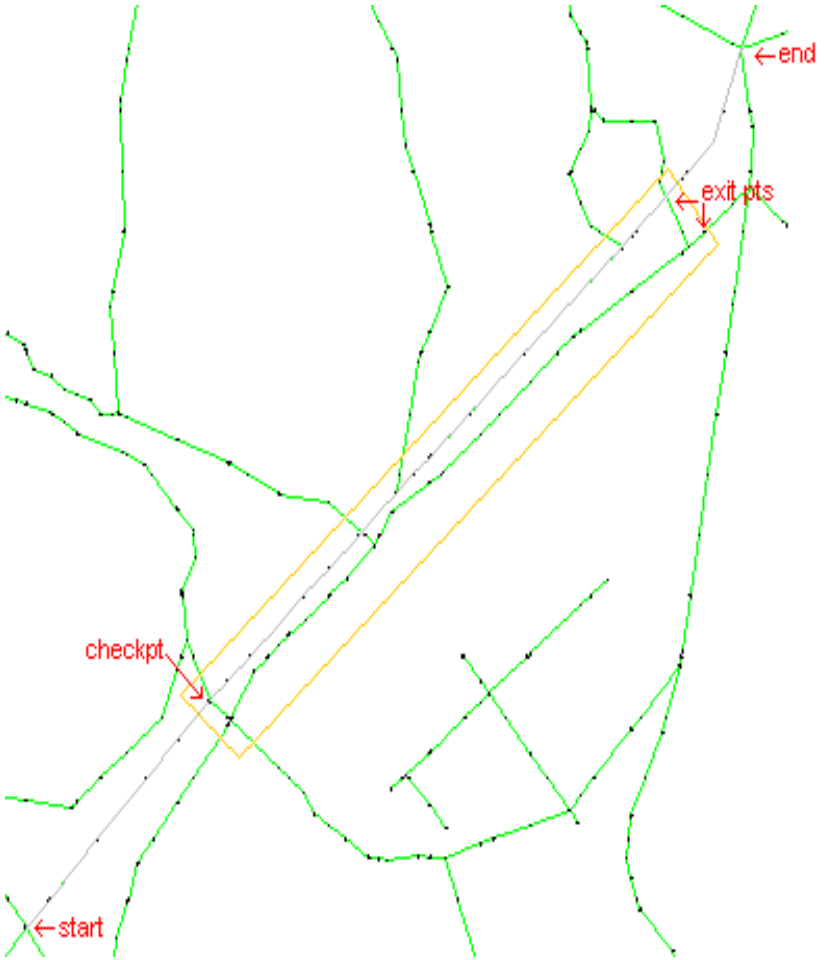
- A no go area currently is a polygonal area that is off limits to the given vehicle.
- Route planning service will not consider any arc wholly, or partially, located within



Axis of Advance

"A line of advance assigned for purposes of control; often a road or a group of roads, or a designated series of locations, extending in the direction of the enemy." (DOD Dictionary of Military Terms at http://www.dtic.mil/doctrine/jel/doddict/)

- **An axis of advance is a polygonal area that encloses one or more roadways.**
- **Vehicle movement is limited to only those arcs and nodes that lie within this polygon**
- **A checkpoint node is designated for the vehicle to enter the axis of advance. This node is the first node that the vehicle visits inside the polygon.**
- **All potential exit nodes are designated as well.**
  1) **An exit node is the last node in the axis of advance that a vehicle may visit before leaving the polygon.**
  2) **When two or more exit nodes are specified, the route planning service will choose the "best" exit that provides the least**



Threat Circle

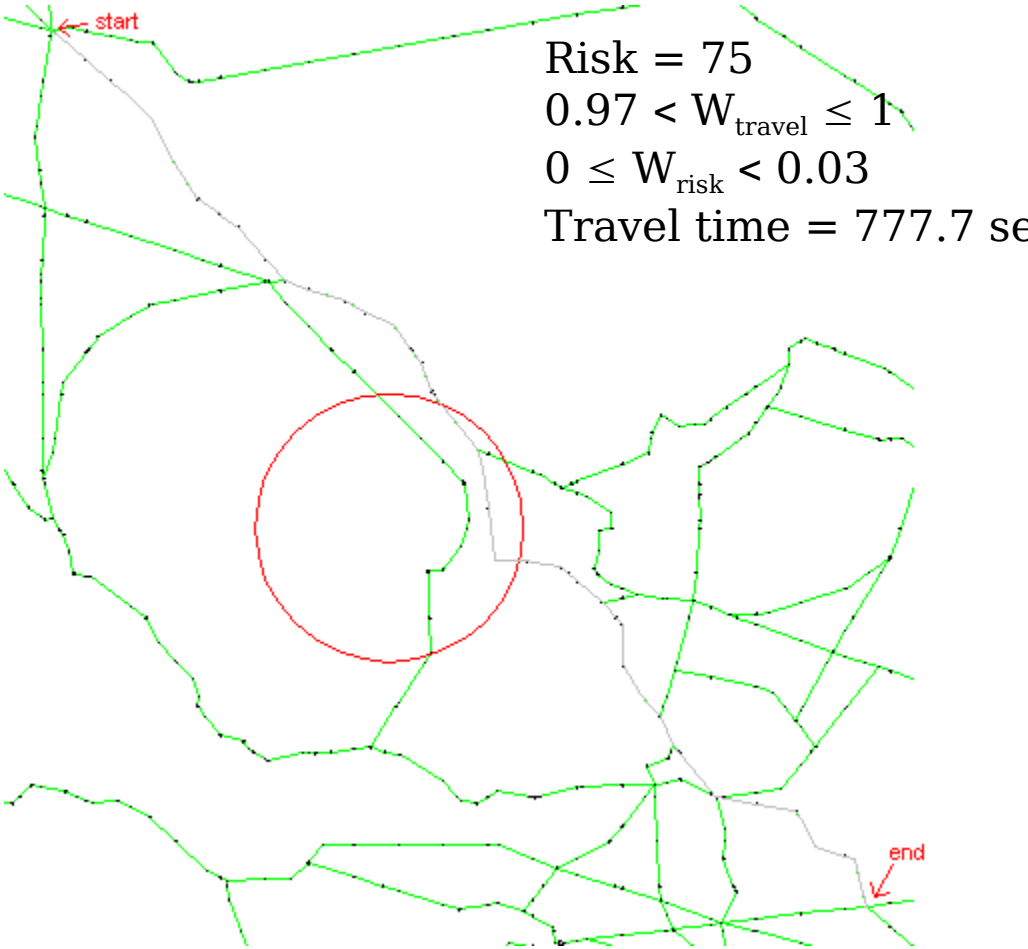
- **A threat circle is defined by a circular area where a threat is perceived to exist.**
- **The area covered by the circle is treated as low-, medium-, or high-level threat relative to the vehicle requesting the route planning service.**
- **A weight is associated with each threat level.**

  1) Larger values indicate that the risk is greater for the vehicle bin type than smaller values.

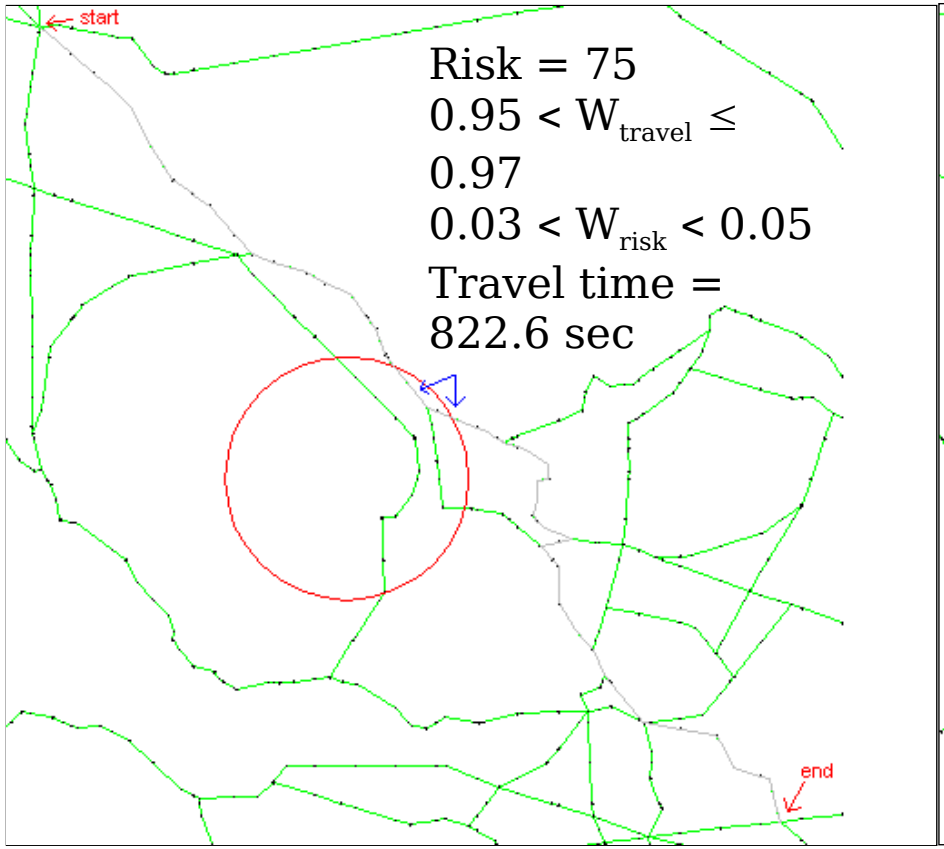
  2) The weight is incorporated into the cost of any arc that lies in the threat circle.

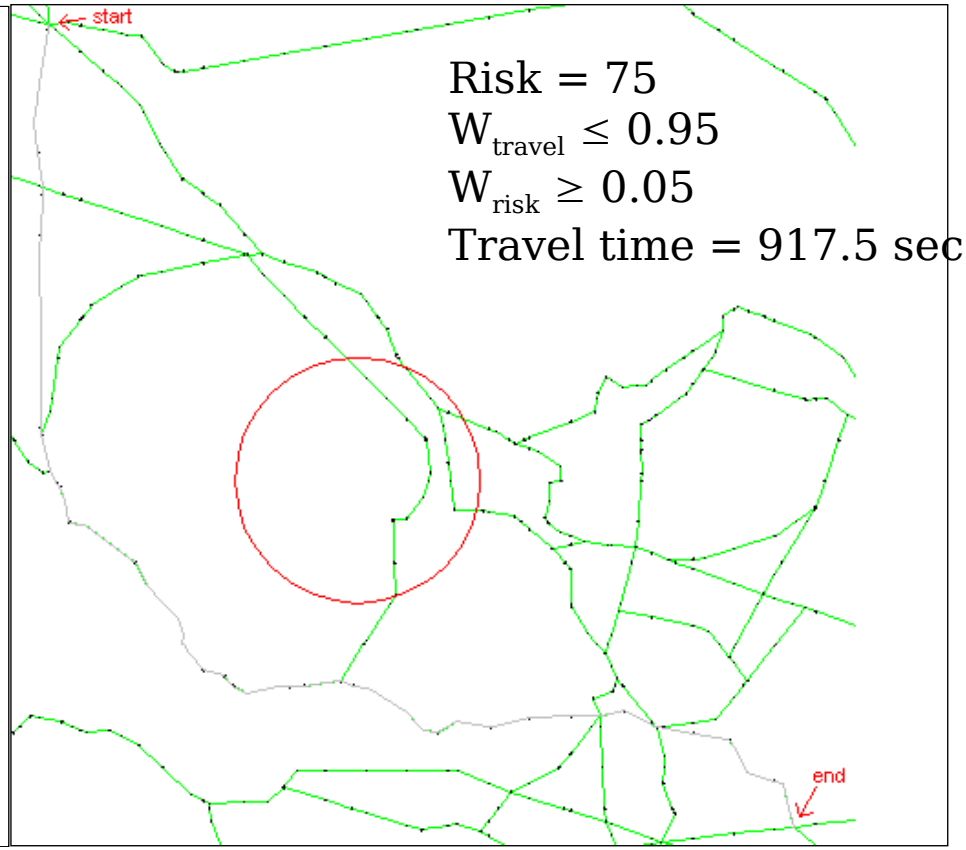
  3) Where an arc is covered by two or more threat circles, the largest weight is used.

  4) Currently the weight is on a scale from 1 to 100.



Threat Circle (cont.)





Turning time delays

Methodology has been developed for the12 Mobility API vehicle bi

Mobility API provides maximum speeds on road arcs:

Travel time from A-B-C $\frac{d_{AB}}{V_{AB}} + \frac{d_{BC}}{V_{BC}}$

Travel time from 1-B-4 $\mathbf{t_{orig}} = \frac{d_{1B}}{V_{AB}} + \frac{d_{B4}}{V_{BC}}$

Turning time delays account for:

- Deceleration into a turn
- Maximum speed in turn
- Acceleration out of a turn

Travel time from 1-2-3-4 =

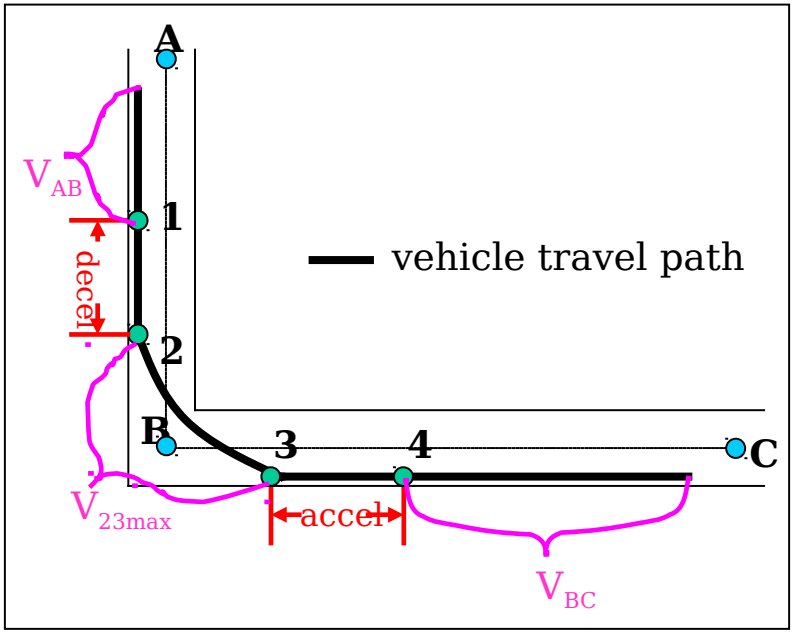
$$\mathbf{t_{new}} = t_{12} + \frac{a_{23}}{V_{23max}} + t_{34}$$

Modification of travel time:

$$t_{delay} = t_{new} - t_{orig}$$

New travel time from A-B-C =

$$\frac{d_{AB}}{V_{AB}} + \mathbf{t_{delay}} + \frac{d_{BC}}{V_{BC}}$$



Case 1		time delays (s)			
		V_{AB}=30mph, V_{BC}=30mph		V_{AB}=35mph, V_{BC}=35mph	
vehicle	V23max (mph)	0% slope	3% slope	0% slope	3% slope
LAV25	21.11	1.35	2.00	2.63	4.43
M113A2	21.91	2.29	13.32	6.28	102.62
M1A1	21.98	1.38	1.97	3.21	5.10
M911M747	19.46	6.27	----	13.34	----
M917	21.28	1.89	5.68	4.04	17.97
M985M989	18.99	3.19	8.76	5.92	22.84
M1084	20.87	1.41	2.00	2.69	4.27
AVLB	22.29	7.75	36.73	****	****
MLRS	22.23	1.56	2.65	3.92	7.73
M1084M1095	20.84	2.09	5.58	4.27	15.72
M985	18.99	2.33	3.98	4.04	8.12
KAW-ATV	22.07	0.79	0.85	1.63	1.84

Turning time delays

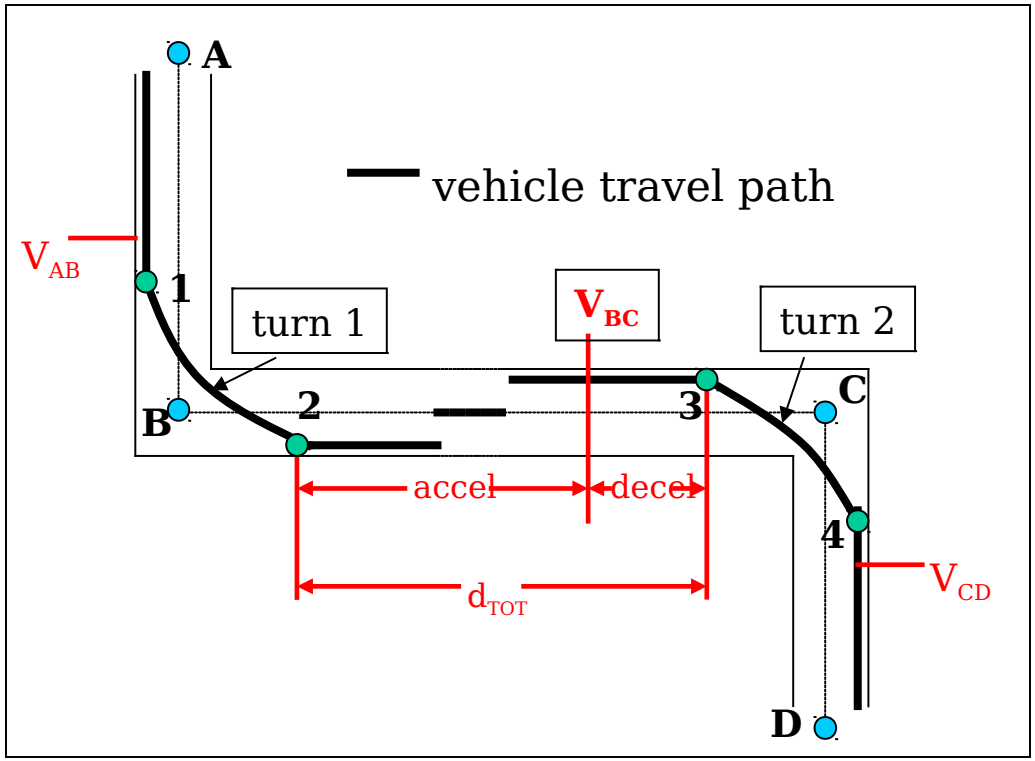
What if a vehicle makes two (or more) turns that are a relatively short distance apart?

Example Problem:

- Vehicle travels from A-B-C-D
- Road arc BC is not of sufficient distance for vehicle to achieve the mobility API prescribed speed V_{BC}.
- Time delays for turns 1 and 2 are a function of V_{BC}

Solution:

V_{BC} is modified (lowered) to accommodate the max acceleration capability of vehicle



Road Ahead: Pathfinder API

Development of an urban route planner is well und

Accomplishments

1) A cost function has been defined for Combat^{XXI}
2) A cost function can be user-defined
3) Constraints modeled:
   a) No go areas
   b) Axis of advance
   c) Threat circles
   d) Turning delays

Under development

1) Methodology for modifying network to bypass obstacles
2) Turn restrictions/detour constraints
3) Integration with the ERC (Combat^{XXI} and OneSAF)